Presented at the 2005 National Symposium on Sensors and Data Fusion

Early Experiments with Ontology-Based Fusion

Date May 2005

Dave McDaniel, J. Wesley Regian
Silver Bullet Solutions, Inc.
2121 Crystal Drive, Suite 708
Arlington, VA 22202
davem@silverbulletinc.com, wes@silverbulletinc.com

Gregory Schaefer
Silver Bullet Solutions, Inc.
4747 Morena Blvd., Suite 350
San Diego, CA 92117
gschaefer@silverbulletinc.com

Abstract

There is a growing sense in the fusion community that an underlying ontology would improve fusion. We explored the idea and reported the results of experimentation to ISIF in 2002 and 2003. In 2004 we have continued with theory development and experimentation. The experimentation described in this paper involved the development of a formal ontology from a data model so that automated processes can reason dynamically, by virtue of onthe formal properties of the ontology relationship types. The experimentation has been for a next generation fusion architecture that is an open architecture in the well-documented sense that adds an ontology layer for further decoupling and coordination of software components. The experimentation has involved rehosting of existing fusion algorithms to operate within the ontology and a publish/subscribe architecture. While the experiments to date have shown how the components can be made to interoperate at the data level, we believe the architecture will ultimately promote or enforce probabilistic interoperability between components providing a fusion open architecture at the probabilistic level.

Keywords: Data Fusion, ontology, semantic modeling, real-time DBMS, embedded DBMS, open architecture, fusion architecture, command and control, IDEF5

1 Introduction

This paper describes research on an open fusion architecture for Airborne Early Warning [1] systems that require extensive automated sensor and data fusion. The goal of this architecture is to provide a foundation for advanced fusion algorithms including non-kinematic level 1 fusion, such as multi-source identification, level 2 and 3 complex assessments, more broadly scoped Situation Awareness and Battle Management information analysis, and level 4 process adaptation [2], [3]; few operational systems provide these types and levels of fusion [4]. Advances in these types and levels of fusion are decreasingly limited by processing speed and power, yet increasingly limited instead by scale, integration, and interoperability issues. For instance, the complexity of the input data for even level 1 target ID processing is staggering -- a priori sources are difficult to manage, to groom for automated processing, and to account for in a mathematically rigorous algorithm set.

We are researching an ontology-based fusion architecture as a way to make it easier to plug in new algorithms that access more types of information. We believe this architecture will support increased automation and higher quality data fusion through enforced integration and integrity of data – thus allowing advanced mechanisms, such as ontology-based inference, as well as the ability to execute multiple kinds of fusion algorithms that interoperate autonomously, yet synergistically. In this paper we describe experiments we have conducted on this architecture.

2 Goals of Ontology-Based Fusion

An ontology is simply a model of the world. In Philosophy, the term refers to a model of the entire world, a systematic account of existence. In artificial intelligence (AI), ontology refers to a model of some part of the world – some domain, or area of interest. In AI, we require an explicit formal specification of how to represent the objects, concepts and other entities that are assumed to exist in some area of interest and the relationships that hold among them. There is a very close correspondence between the military concept of a Common Operating Picture [5] and the concept of ontology in AI. They both refer to the way we understand and thus reason about a situation. Military commanders, like intelligent agents, reason better if they have a shared, accurate, and complete understanding of the situation on the ground. In AI systems, the only things that "exist" are things which can be represented. When the knowledge about a domain is represented in a declarative language, the set of objects that can be represented is called the universe of discourse. We can describe the ontology of a program by defining a set of representational terms. Definitions associate the names of entities in the universe of discourse (e.g. classes, relations, functions, or other objects) with human-readable text describing what the names mean and formal axioms that constrain the interpretation and well-formed use of these terms. Formally, an ontology is the statement of a logical theory.

Ontologies are of interest to fusion researchers and developers [6] because it defines the rules by which the fusion algorithms must operate. This can support proper behavior of a specific fusion algorithm and can enable the proper interaction of multiple algorithms. It can also serve as a data exchange conduit between algorithms and data sources – sensors, a-priori, and others. There seems to be many diverse perceived benefits of an ontology amongst fusion researchers and developers. The benefits we understand are shown in Table 1. The ones that have guided our work can be categorized as to the time-frame of believed benefit. That is, some benefits we believe accrue early to the fusion system once the technology is injected; others later, and others much further downstream as the ontology matures and as fusion algorithms are developed that can exploit it. While beyond the scope of this paper to discuss each of these in detail, some brief discussions of the believed benefits follows.

- *Inadequate prior knowledge utilization*: Good analysts make use of enormous amounts of prior knowledge when trying to understand current information from diverse sensors and sources. In order for automated systems to support utilization of prior knowledge, that information must be available to the system in a standard, machine-readable format

- *Information scope brittleness and shortfalls*. Most fusion algorithms, and virtually all applied fusion algorithms, deal with a narrow scope of information. Powerful inferential reasoning generally builds on widely divergent types of information that combine to generate a consistent understanding of the situation.

- *Insufficient exploitation of weak and indirect evidence*. In many cases, an analyst's assessment of a situation is not based on one single definitive piece of information, but on an overarching assessment of many smaller indicators, any one of which would be inconclusive – and many of which may be quite indirect. In order for automated systems to exploit weak and indirect evidence, they must recognize and accumulate relevant bits of information and update assessed probabilities based on the evolving weight of evidence. Inference network techniques provide a way to cope with massive amounts of interrelated variables via the Markov construction of the network that explicates probabilistic and causal dependencies. Between the class structure, properties, and inter-relationships of the ontology and the casual and correlated representations in the inference network, there is much opportunity for advances in intelligent computing.

Table 1. Some Possible Benefits of Ontologies in Fusion

Benefit	Time-Frame
Common data exchange amongst fusion algorithms and data sources	near
Implementation of referential integrity, datatype, and valid values	near
Maintenance of multiple hypotheses in a uniform manner	mid
Pedigree source and temporal tracking	mid
Software open-architecture publish / subscribe invocation method	mid
Relationship type rules	mid
Autonomous interoperation of fusion algorithms	mid
Inadequate prior knowledge utilization	mid
Weak evidence accumulation and interaction.	far
Automated reasoning in support of the fusion process	far
Information scope brittleness and shortfalls	far

3 Ontology Modeling

For this project, we needed to model an ontology for command and control that can be implemented and executed in a computer system, that is scalable to the broad scope of command and control, and that is based on domain expertise . We needed it to be implementable in a computer system in the sense that the concepts, rules, and other aspects of the ontology are imparted to the computer program in a way that it executes upon them. To contrast, in a typical database schema, the definitions of entities, attributes, relationships, and valid values are for human discussion and understanding only; the DBMS receives and operates only upon the datatypes, the relationship cardinalites, NULL rules, valid value lists, and so forth. That is, very little of what we perceive as the 'semantics' is actually executable by the machine. Our aim in the ontology modeling is to impart additional semantics to the machine in a way that is executable. In this section we briefly review how we are going about this. There are some more details in [7].

3.1 Object-Oriented Entity-Relationship Models

One of the more detailed and rigorous entity-relationship modeling standards, IDEF1X [8], draws primarily on three foundations, specifically Chen's Entity Relationship (ER) model [9] Codd's Relational model [10], and Smith's Aggregation/Generalization model [11]. The original models by Chen and Codd were shown to be mathematically correct and implementable. The extensions by Smith support abstraction modeling akin to object class hierarchies in object-oriented database designs [12] Abstraction is essential in ontology modeling. It is simple and natural for a human fusion expert to see corresponding elements from intelligence and track databases as imperfect reflections of a single entity – but viewed from two perspectives. It is not straightforward for traditional fusion software to make this sort of abstraction-derived inference. The lack of an abstraction layer can cause data integrity problems, allow inconsistent business rules, and overly complex processing logic. Multi-tiered relationships allow explicit representation of abstraction.

An example of the use of the abstraction modeling in IDEF1X is the C2 Information Exchange Data Model (C2IEDM), a NATO standard data model for command and control [13]. A very high-level overview we developed of C2IEDM is shown in Figure 1. Even this overview figure shows extensive object class hierarchies. These provide compactness and extensibility -- compactness because common properties do not have to be repeated and extensibility because new data elements can add new properties to those from their superclass. We think of models such as this as Object-Oriented Entity-Relationship (OOER) models that combine the strengths of both modeling techniques.

3.1.1 Data Model Extensions for Fusion

We are extending the original C2IEDM model in several ways to allow multiple hypotheses consideration, increased quantitative precision, full temporal specification, uncertainty representation, and pedigree as described

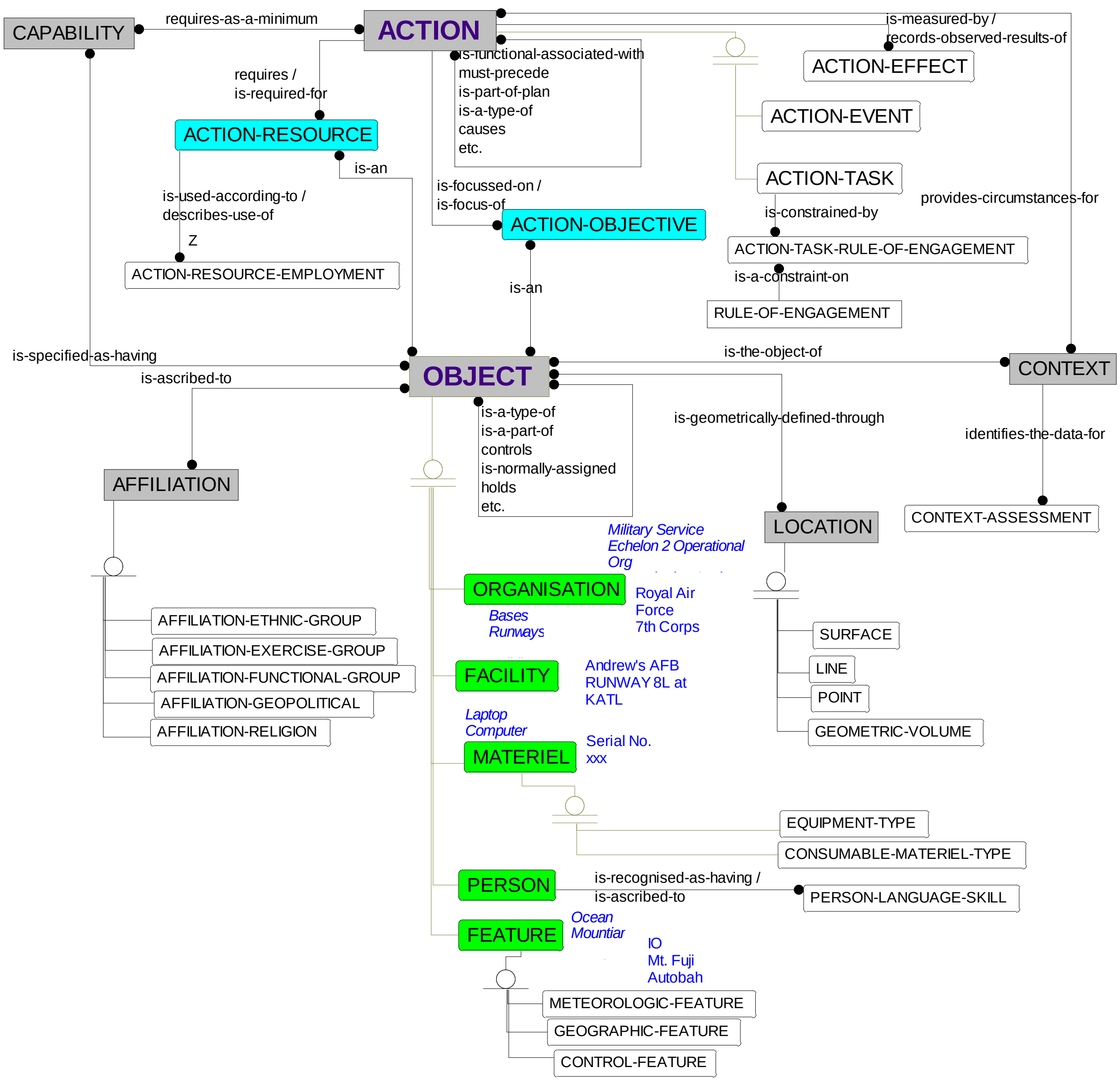

Figure 1. Our C2IEDM Conceptual Overview

briefly in the following subparagraphs.

3.1.2 Uncertainty handling

We explicitly represent uncertainty to enable machine reasoning about information, beliefs, and uncertainty parameters derived from various fusion engines. Uniquely, this approach addresses one of the most difficult issues in achieving situation awareness, that is, real-time fusion of (sometimes inconsistent) information from disparate sensors and sources.

3.1.3 Precision

The model did not support precision tracking and timing so it was necessary to add a 9-state kinematics, and an additional attribute for sub-second time maintenance. We have not yet added body orientation (e.g., direction cosines) and body axis displacements and rates (e.g., roll, pitch, yaw and their rates) but these would fit well as amplifiers on RELATIVE-POINT in the same manner as kinematic-state. These data elements will be important for ballistic missile tracking. It will probably be necessary to add uncertainty representations for orientation and navigation because of the importance of accurate and precise estimation for this type of application.

3.1.4 Temporal and Multi-Belief Dimensions

The C2IEDM has a data structure for maintenance of data source called REPORTING-DATA. In order to support both multi-beliefs (e.g., local and remote tracking) and maintenance of track history all that had to be done was move the REPORTING-DATA foreign key to be part of the primary key.

3.1.5 Pedigree

We extended the model to enable explicit representation of information pedigree - including source, certainty, and lineage. These extensions incorporate and build on parts of a metadata model (metamodel) developed by the Joint Strike Fighter (JSF) program to manage information products supporting modeling and simulation [14].

3.2 Relationship Modeling (IDEF5)

An important difference between data models and ontologies is the formalization of the specified relations. In data models, the relations (is-specified-by, contains, is-part-of, etc.) are not formally specified. They are intended to be understood by humans and are not sufficiently defined so as to be understood by machines for purposes of inference routine execution. In ontologies, the relations are formally and axiomatically specified. They provide the basis for machines to generate inferences based on First Order Predicate and Modal Logics.

IDEF5 [15] draws on additional foundations, including Semantic Nets [16, 17], Situation Theory [18], Set Theory, FOPL [19], and the Modal Logics [20]. It models concepts and conceptual relations as ontologies, and thus provides an abstract level of representation for describing a domain in a manner which closely reflects the human conceptualization of that domain. Ontologies are defined under IDEF5 in 4 steps. First, we provide an inventory of the kind of objects that exist within a given domain according to the best sources of information. Second, for each kind of object we provide a description of the properties that are common to all and only instances of that kind. Third, we characterize the particular objects that in fact instantiate the kinds within the domain. Finally, we provide an inventory of the relations that exist within the domain between and within kinds of objects.

For example, “Part-of,” a commonly used relation, is inherently too ambiguous for machine processing. Under IDEF5, the meronymic relation (part-of) is formally defined as 6 different kinds of part-of relations. These include Physical-Part-Of (place-within, component-of, stuff-of, portion-of) and Conceptual-Part-Of (member-of, activity-within). Each of the six is associated with a set of axioms that include FOP and Modal logical operations that can be performed on entities that are related. Thus a machine can conduct logical operations on meronymic relations.

IDEF5 employs two languages. The schematic language, using graphical symbols to express ontological information, is very similar to IDEF1X. The elaboration language enables the expression of axioms. The core of the IDEF5 Elaboration Language is based on the KIF [21], which includes a predefined library of relation constants. The elaboration language contains all definitions and characterizing axioms in a structured text language with the full expressive power of first-order logic and set theory. It can express almost any condition, relation, or fact needed to express any given kind of thing, property, relation, or process found in a domain. The syntax of the language uses a prefix notation and parentheses to delimit expressions. The alphabet for the language consists of the standard alphanumeric and punctuation characters.

The notion of a word is taken as a primitive of the IDEF5 Elaboration Language. A sentence in the language is composed of operators, constants, and, possibly, variables. The IDEF5 Elaboration Language supports three types of expressions: terms, definitions, and sentences. Terms are used to denote objects. A definition is a type of expression used to define an individual, relation, or function constant. A definition can be complete or partial. A sentence in the IDEF5 Elaboration Language expresses some fact about the constants in the ontology. There are seven types of sentences in the language. There are eleven categories of IDEF5-specific sentences, each category corresponding to a concept in the IDEF5 method. The IDEF5 elaboration language also includes relation constants, which are predefined in the language and, hence, can be used in sentences.

We are new strengthening our fusion ontology model from IDEF1X specification to IDEF5 specification - essentially turning it into a fusion ontology modeling language to enable machine understanding and automated reasoning.

4 Experiments with Fusion Algorithms

4.1 Real-Time Database

We implemented the model in a real-time database for data storage and retrieval. For these experiments we used TimesTen™ [22]. It provides database software for real-time event processing – a fundamental requirement of time-critical applications used in command and control systems. TimesTen™ is a re-write of ANSI SQL to optimize for RAM versus disk-based virtual memory access. It supports open architecture with ANSI SQL compliance, binding to common programming tools, and an interface to offline DBMS’ such as Oracle. We reported some early experimental results on TimeTen’s performance for real-time command and control in [23].

4.2 Tracking Filters

We conducted two tracking filter experiments. In both of these the subset of the ontology used was very small, the purpose of the experiments being to see what was involved in conforming existing software to the architecture. The ontology fragment implemented in the embedded DBMS is as shown in Figure 2. In these experiments, a simple simulator generates radar processor output reports in the form of Range and Azimuth data. As these records are added to the RECEIVER-OUTPUT table, the tracker filters are triggered to execute. Note that in both the RECEIVER-OUTPUT and KINEMATIC-STATE tables, records are never over written but are only added, by virtue of the design which placed time as an identifying key attribute, part of a composite key. Although many fusion systems today do not maintain this temporal dimension, there are benefits that may be worth considering now that computing resources allow the option. For the purposes of the experiments, the temporal dimension allowed us to look at the tables after the scenario runs.

Because the code we had had been previously used in a test environment to develop the existing actual code, it had file loaders and outputters at the beginning and end that we had to delete. Then we had to build a 'wrapper' that would respond to the DBMS trigger and translate the data from the ontology to the format the legacy code required. This architectural "framework" [24] is shown in Figure 3. The results were that this worked; the simulator filled the RECEIVER-OUTPUT table and as each record was added, it triggered the wrapper which executed the filter's code which the wrapper then formatted into the KINEMATIC\-STATE and KINEMATIC-COVARIANCE tables.

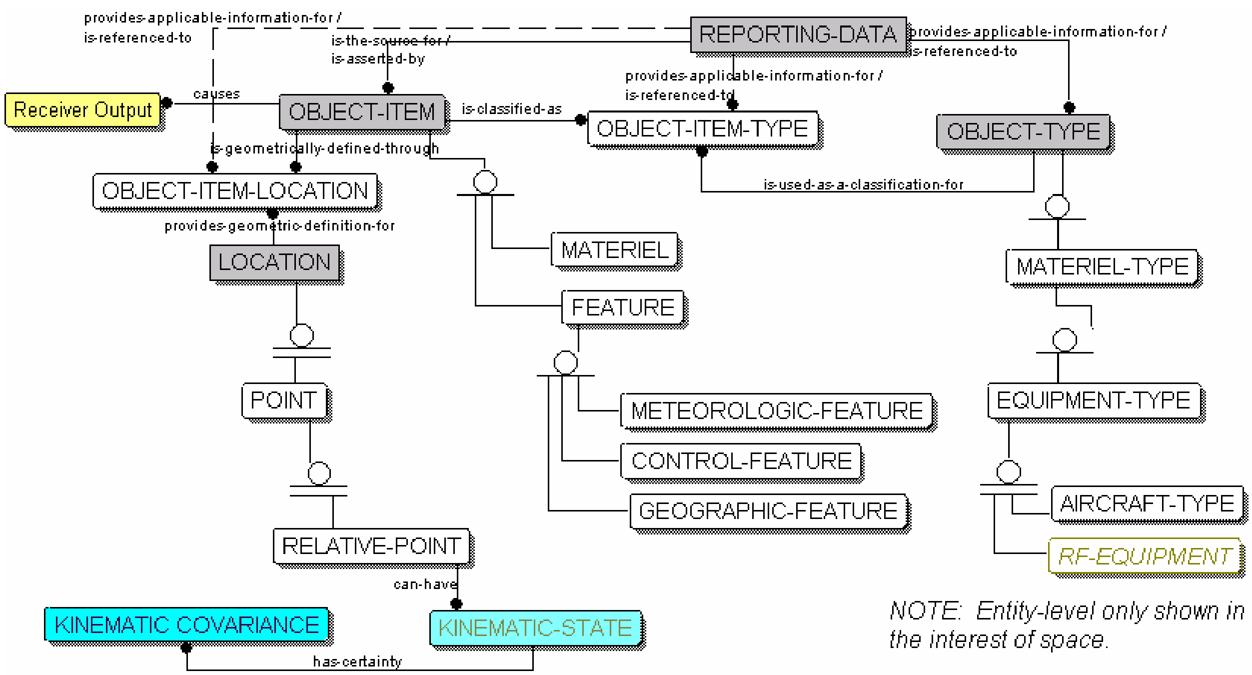

Figure 2. Ontology Fragment Used for Tracking Filter Experiments

4.3 Association

The ontology fragment used for these experiments is shown in Figure 4. The assignment matrix is maintained in the OBJECT-ITEM-ASSOCIATION structure which supports showing a relationship between two OBJECT-ITEM instances, in this case with an association type code of “correlated”. The association confidence value, the same attribute used for hypothesis confidence throughout the ontology, was used to maintain the cost or probability, depending on the algorithm. In a final architecture, a standard confidence value would be defined and all interacting algorithms would have to have their wrappers do any de-biasing or translation of correlation scores, costs, etc. to the confidence value standard.

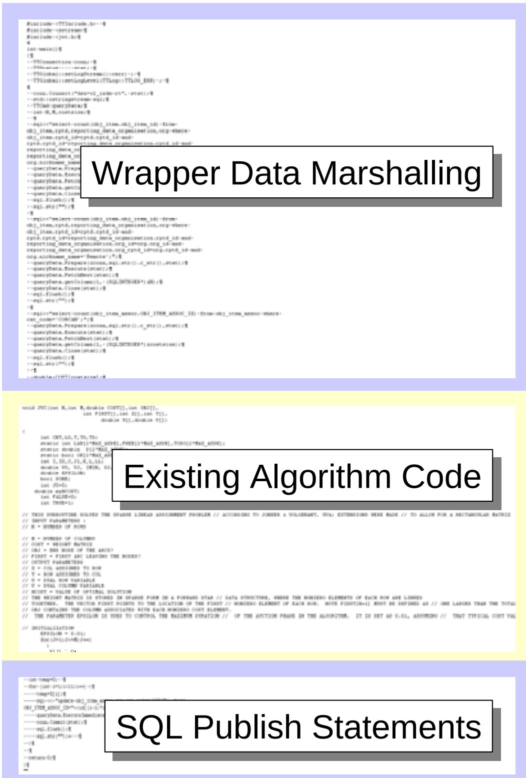

Figure 3. Fusion Architecture Wrapper Framework

The JVC [25] code provided by Dr. Oliver Drummond (many years ago) had file readers and outputters on the ends that had to be eliminated. However, from that point on, the conformance was simply a matter of data translation since all the persistent data was maintained in the ontology. A data translation specification was developed, to go back and forth between the application-dependent data structures and the ontology fragment. Note that the sparse matrix handling in the original code is handled very elegantly by the associative data structure in the ontology and yet achieves the same goal, that of storing only the sparse feasible costs and not the entire matrix. The difference is that the DBMS can lookup the sparse entries given the indices because it uses state of the art hashing and indexing to lookup all data. In effect, the DBMS has sparse matrix handling as a special case of its more general handling of all sorts of lookups.

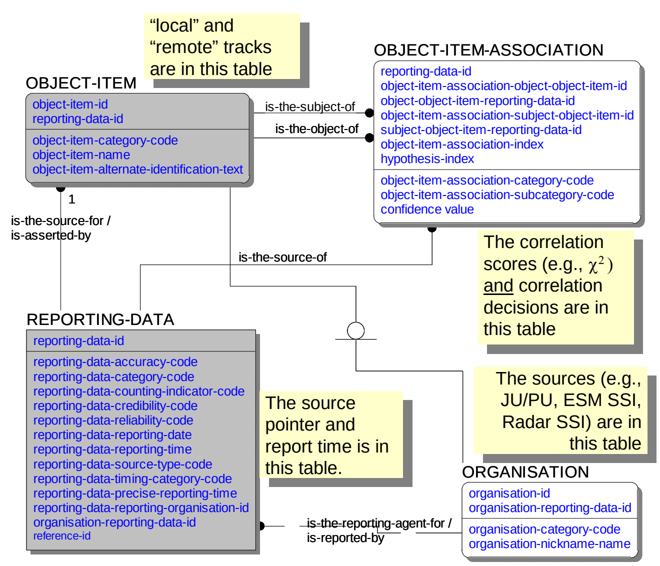

Figure 4. Ontology Fragment for Assignment Algorithm Experiments

The cost matrix inputs were generated using the two trackers in the prior experiments. The RECEIVER-OUTPUTs had bias applied to simulate navigation and sensor errors typical of gridlock and sensor data registration seen in the field, that is, lat, long, and azimuth offsets and a multiplicative range scale factor. In addition, a probability of detection was applied based on a hypothetical range to target and a random function. As well, the biases were randomly varied. Again, this experiment was a success, as the code was conformed to run against the ontology.

At this point it is worth again noting the temporal dimension of the ontology since now it allows for backtracking of reasons-why for correlation decisions. This can be very important in aberrancy removal. While the goal is to have as few correlation aberrancies as possible, they are inevitable. The temporal dimension could provide a more elegant way to handle than some of the complex aberrancy removal logics we have seen. Because of the known complexity, some systems make a design decision to not attempt to re-do fusion decisions and estimates and instead just tolerate data contaminations known to be present when decorrelations or change correlation decisions occur.

4.4 Classification

The purpose of this experiment is to see what is involved in inferring across several nodes in the ontology to perform ESM classification. The fusion process is that an ESM track is provided with basic classical parameters: From this information, an estimate of the type of platform is desired. The fusion inference is find the most likely type of platform to have caused this signal measurement to occur, the causality starting with the platform which causes its radar to operate which, in turn, causes the signal which, in turn, causes the ESM receiver to excite which causes the measurement to be sent to the fusion processor. Note that this causal analyses and modeling invites a more careful interpretation of the data, e.g., the biases in the measurement caused by our receiver [26]. However, for this experiment, we simplified and used the model subview shown in Figure 5. The full inference algorithm code involves consideration of factors such as antenna gain and beamwiths, effective radiated power, and propagation conditions.

For this experiment we created an a-priori database that approximated the ambiguity characteristics of a real classification system; around 40,000 modes, 4,000 emitters, and 70,000 platforms. This experiment evidenced some challenges to the architecture in that, unlike the previous experiments described, marshalling the data was impractical due to the massive quantities that would have to be repeatedly marshaled and loaded into complex data structures. That is because there can be many classification candidates input to the algorithms if there is much ambiguity in the input report. The solution was to instead create an object layer that the algorithms could interact with directly that would map to the ontology. Its mechanism is based on light-weight classes bound to a database [27]. Triggers ensure the persistent objects remain in synchronization efficiently. Easily customizable, it can be adapted to any data structure. This architecture presents yet another challenge and that is error handling when the algorithm tries to do something the broker considers illegal. While correct from an overall system point of view to maintain the integrity of the database, it does mean error handling has to be added to legacy code. The tension between database integrity and easy access is a long standing challenge in the use of DBMS and other brokers in embedded systems and our approach, one of many, is simply another step towards what may someday be a more standard one.

5 Future Work

The next phase of this project is to take the initial research further into implementation. This will be with experimentation with existing and new technology software elements running against the next generation architecture in the avionics integration lab. We will be conducting experiments with software elements planned for

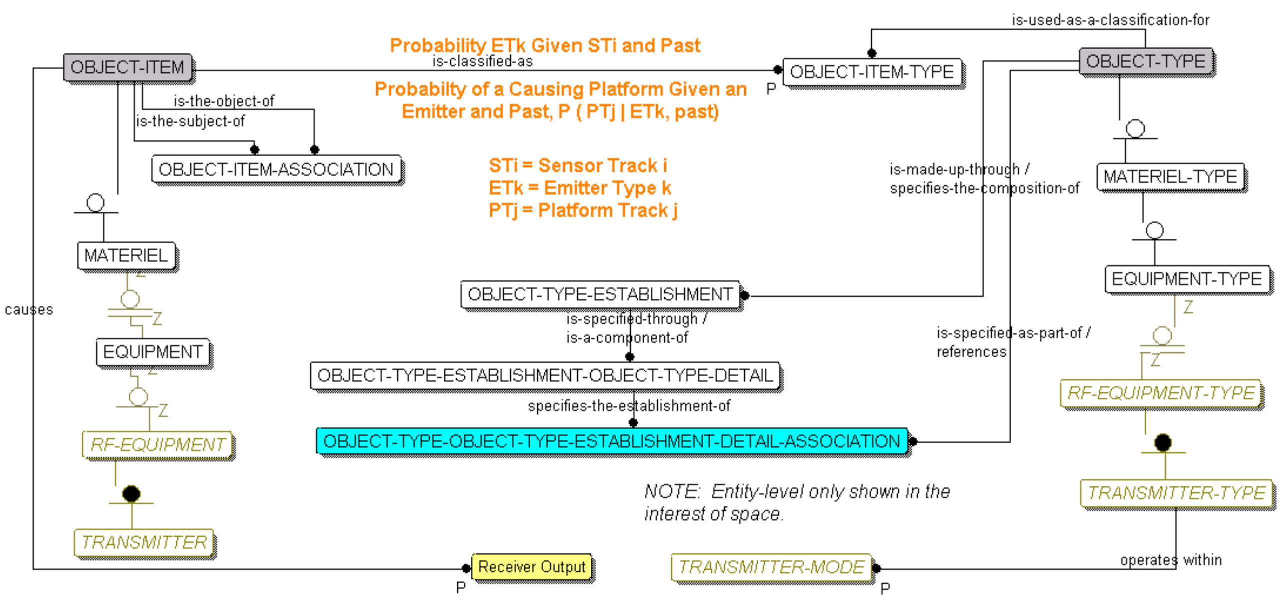

Figure 5. Classification Experiment Ontology Fragment

the aircraft to mature the fusion architecture in preparation for full scale development. There will be ontology extensions, there will be adjustments to the publish / subscribe mechanisms, and implementation in mission system at the integration lab. Also, as early results from the mathematical formulation of the ontology-based fusion theory are nearing maturity, we plan on performing experimentation in conjunction with the aircraft's mission system. The focus of our efforts with the aircraft system to date, and thus the focus of this paper, has been on JDL fusion levels 0 and 1. It is our intention, however, to provide a foundation for machine understanding and automation support for fusion levels 2+ in the future. For example, we are developing a prototype ontology model to begin experimentation on supporting tactical situation awareness.

6 Conclusions

The results have been very positive. Fusion software from actual legacy systems has been conformed to operate under a publish / subscribe triggering mechanism embedded in an ontology derived from a highly interoperable object-oriented model implemented in an embedded DBMS. At the same time we have been able to push the envelope on formalizing a way to use the ontology for more advanced fusion inference foreseeable in the future.

7 References

[1] Sponsored by the U.S. Naval Air Systems Command under Small Business Innovative Research contract N00421-04-P-0747.

[2] E. Waltz, J. Llinas, Multisensor Data Fusion, Artech House, Inc., Norwood, MA, 1990.

[3] A. N. Steinberg, C. L. Bowman, and F. E. White, Revisions to the JDL Data Fusion Model, Proceedings of SPIE AeroSense (Sensor Fusion: Architectures, Algorithms and Applications III), page 430-441, Orlando, Florida, 1999.

[4] M. Nichols, "A Survey of Multisensor Data Fusion Systems", in Handbook of Multisensor Data Fusion, ed. By Hall, D., and Llinas, J., CRC Press, 2001.

[5] Defense Information Systems Agency GCCS home page, www.disa.mil/gccs/

[6] see the ontology track in Proceedings of the Fifth International Conference on Information Fusion, International Society of Information Fusion, Sunnyvale, CA, 2002.

[7] D. McDaniel, "Multi-Hypothesis Database for Large-Scale Data Fusion", Proceedings of the Fifth International Conference on Information Fusion, International Society of Information Fusion, Sunnyvale, CA, 2002.

[8] R. Mayer, M. Painter, and P. deWitte, "IDEF Family of Methods for Concurrent Engineering and Business Re-engineering Applications", Knowledge Based Systems, Inc., 1992, www.idef.com/Downloads/pdf/IDEFFAMI.pdf

[9] P. P-S. Chen, `The Entity Relationship Model - Towards a Unified View of Data', ACM Transactions on Database Systems, Vol. 1, No. 1, March 1976, pp. 9-36.

[10] E. F. Codd, CACM V13 #6, pp. 377-87. "A Relational Model of Data for Large Shared Data Banks" June, 1970.

[11] J. M. Smith, and D. C. P. Smith, Database Abstractions: Aggregation and Generalization. ACM Transactions on Database Systems, 2(2): 105--133, 1977.

[12] G. Saake, S. Conrad, I. Schmitt, Turker, "Object-Oriented Database Design: What is the Difference with Relational Database Design?", in Object World Frankfurt '95, Conference Notes, R. Zicari, editor.

[13] Multilateral Interoperability Program Specification, www.mip-site.org/MIP_Specifications/Baseline_2.0/C2IEDM-C2_Information_Exchange_Data_Model/

[14] R. Scrudder, W. Graves, T. Teigen, C. Johnson, S. Hix, J. Hollenbach, "The Critical Role of Metadata in JSF Development", www.msiac.dmso.mil/sba_documents/JSF%20Metadata%2003F-SIW.pdf

[15] IDEF 5 Method Report, Knowledge Based Systems, Inc., 1994, www.idef.com/Downloads/pdf/Idef5.pdf

[16] A. M. Collins, & E. F. Loftus, *A spreading activation theory of semantic processing*. Psychological Review, 82, 407-428, 1975.

[17] J. F. Sowa, Semantic Networks, jfsowa.com/pubs/semnet.htm

[18] J. Barwise, *Scenes and Other Situations*, The Journal of Philosophy, vol. 77, pp. 369-97, 1981.

[19] G. Hunter, Metalogic: An Introduction to the Metatheory of Standard First-Order Logic; Macmillan and University of California Press, 1971.

[20] R. Feys, Modal Logics. (ed. J. Dopp). Louvain: Coll. de Logique Math. series B, 1965.

[21] M. R. Genesereth, Knowledge Interchange Format - draft proposed American National Standard (dpANS). NCITS.T2/98-004. Online Paper as of 1999-09-04, logic.stanford.edu/kif/dpans.html

[22] TimesTen™ home page, www.timesten.com

[23] D. McDaniel, and G. Schaefer,"Real-Time DBMS for Data Fusion", Proceedings of the Sixth International Conference on Information Fusion, International Society of Information Fusion, Sunnyvale, CA, 2003.

[24] Open Architecture Volume 4, Open Architecture Computing Environment Technologies and Standards, Version 1.0 ; May 2004.

[25] R. Jonker, A. Volgenant, "A Shortest Augmenting Path Algorithm for Dense and Sparse Assignment Problems", Computing, Vol. 38, 1987.

[26] D. McDaniel, Electronic Warfare IDentification (EWID) Small Business Innovative Research Final Report, Space and Naval Warfare Systems Command, 1996.

[27] Java Data Objects developer community page, www.jdocentral.com)